To-Do List Application – SDLC Assignment

Submitted by: Khushi Manoj Kasliwal

Project Title: To-Do List Application

Course: Software Engineering / SDLC

1. Project Description

The To-Do List Application is a simple, user-friendly web-based productivity tool designed to help users manage their daily tasks. The primary functionalities include adding new tasks and deleting completed or irrelevant tasks. The application is designed with clean and responsive UI principles to provide a pleasant user experience across devices.

The purpose of this project is to implement the software development life cycle (SDLC) methodologies in a practical context. Each phase—planning, analysis, design, implementation, testing, and maintenance—was followed and documented during the project lifecycle.

2. Tools and Technologies Used

- HTML – For structuring the content of the web page.
- CSS – For styling the interface and ensuring a responsive design.
- JavaScript – For dynamic interaction such as DOM manipulation, input field clearing, and future LocalStorage integration.
- Code Editor: Visual Studio Code (VS Code)
- Web Browsers: Google Chrome

3. Product Backlog

ID	User Story	Priority	Status
PB1	As a user, I want to add a task to my to-do list	High	✅ Done
PB2	As a user, I want to delete a task from my list	High	✅ Done
PB3	As a user, I want the UI to be responsive and clean	Medium	✅ Done
PB4	As a user, I want tasks to be stored on refresh (LocalStorage)	Low	❌ Not Done

4. Sprint Planning

Sprint Goal:
To deliver the core functionality of the To-Do List app—task creation and deletion—within a simple and visually appealing user interface.

Selected User Stories for Sprint:

- PB1: Task Addition
- PB2: Task Deletion
- PB3: Responsive and styled UI

5. Daily Scrum

Day	What was done	What will be done	Blockers
Day 1	Basic structure using HTML and CSS	Implement task addition using JavaScript	None
Day 2	Implemented task deletion and styled UI	Testing, cleanup, and screenshots	None

6. Sprint Review

Completed Features:

- A fully functional input field with an "Add Task" button.
- Dynamically generated list of tasks with "Delete" functionality.
- Also generated a “Mark as Done” functionality and a undo option.
- Clean layout compatible with multiple screen sizes.

---

7. Sprint Retrospective

What went well	What didn’t go well	Improvements
UI was completed quickly	Lack of persistent task storage	Plan integration of LocalStorage or backend
Core functions worked smoothly	Minor alignment issues initially	Consider code reusability and modularity

---

8. User Stories

- US1: As a user, I want to add new tasks so I can manage my daily work.
- US2: As a user, I want to delete tasks that are no longer needed.
- US3: As a user, I want the app to look clean and work on any device.

---

9. Impediment Log

Date	Impediment	Resolution
Day 1	Flexbox alignment issues	Fixed using justify-content and align-items
Day 2	Input field not clearing after task added	Reset the input using JavaScript

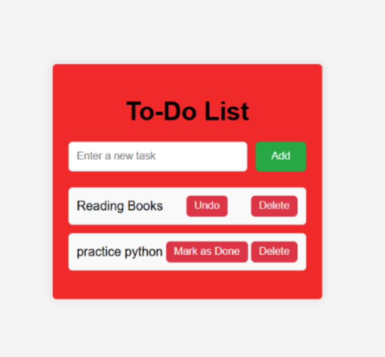
---

10. Definition of Done (DoD)

- ✅ Task can be added using the input form
- ✅ Task can be deleted from the task list
- ✅ Responsive and clean UI layout
- ✅ No JavaScript console errors
- ✅ Works across modern browsers
- ❌ Tasks not yet stored after refresh (Future improvement)

---

11. Screenshot of Application



---

12. GitHub Repository

https://github.com/kkasliwal05/To_do_list

---

13. Conclusion

This project provided hands-on experience in the software development life cycle, from conceptualization to implementation. It helped reinforce principles such as planning sprints, writing user stories, and focusing on the user experience.

Though simple, the To-Do List Application reflects real-world software engineering practices, and its scope can be expanded in the future to include persistent storage, task editing, and categorization.